Comm. in Asteroseismology
2006, in preparation

SigSpec 1.2 User's Manual

P. Reegen[1]

[1] Institut für Astronomie, Türkenschanzstrasse 17, 1180 Vienna, Austria
reegen@astro.univie.ac.at

Abstract

SigSpec (abbreviation of 'significance spectrum') is a program that computes the significance levels for the amplitude spectrum of a time series at arbitrarily given sampling. It is based on an analytical solution for the Probability Density Function (PDF) of an amplitude level, including dependence on frequency and phase. Using a time series dataset as input, SigSpec produces both an amplitude and a significance spectrum. Furthermore, the frequency, amplitude and phase associated to maximum significance in the considered frequency range are computed. SigSpec relies on the comparison of the given time series to white noise, i. e. noise without any correlation between consecutive data points. An optional iteration including consecutive prewhitening is provided to determine a whole set of signal components at once, which makes the program a powerful tool also for multi-frequency analysis. Time-resolved significance analysis using a set of intervals of the time series is supported to investigate the development of eigenfrequencies over the observation time. Furthermore, an extension is available to perform the SigSpec analysis for multiple time series input files simultaneously.

What is SigSpec?

SigSpec (abbreviation of 'significance spectrum') is a program that computes a significance spectrum for a time series. It evaluates the *Probability Density Function (PDF)* of a given amplitude level analytically, making use of the theoretical concept introduced by Reegen (2005, 2006). Instead of the frequently used *Cumulative Distribution Function (CDF)* or *False-Alarm Probability*

Φ_{FA}, respectively (e. g. Scargle, 1982), SigSpec calculates the *significance* of an amplitude A by

$$\mathrm{sig}\,(A) := -\log\left[\Phi_{\mathrm{FA}}\,(A)\right]\,. \tag{1}$$

E. g., a significance equal to 8 indicates that the considered amplitude level is due to noise in one out of 10^8 cases. A signal-to-noise ratio of 4 (Breger et al. 1993) is – on average – assigned to a significance of 5.4575. Hence consistency with the S/N = 4 citerion is obtained by relying on signal only if the significance exceeds this value.

SigSpec performs an iterative process consisting of four steps:

1. computation of the significance spectrum,
2. exact determination of the peak with maximum significance,
3. least-squares fit of amplitude and phase for the frequency of maximum significance,
4. prewhitening of the sinusoidal component with maximum significance.

By default, there are three types of output (see "Default Output", p. 7):

1. a Fourier and significance spectrum for each step of prewhitening,
2. a time series representing the residuals after each prewhitening step,
3. a result file, which contains a list of prewhitened signal components.

The amount of spectra and/or residuals to save may be controlled by the user.

SigSpec can produce additional files containing

1. a spectral window for the given time series,
2. a significance profile containing the parameters $\alpha_0\,(\omega)$, $\beta_0\,(\omega)$, $\theta_0\,(\omega)$ determining the ellipticity of noise rms error in the frequency domain, referring to Reegen (2006),
3. a preview of the significance analysis,
4. a Sock Diagram.

For further details, please refer to "Specials", p. 9. These options are deactivated by default.

The program also supports windowed significance analysis, i. e. the subdivision of a time series into a set of intervals and the separate analysis of all these

parts in order to monitor eigenfrequency changes with time. This method will furtherin be called time-resolved analysis. In this case, the output is somewhat richer, as described in Section "Time-resolved Analysis" (p. 17).

The prewhitening loop stops, if no significance level above a pre-defined limit is found. As described in "Program termination", p. 7, there are three different criteria that may be applied to determine the rules for program termination:

1. the number of iterations in the prewhitening sequence,
2. a lower significance limit for the highest peak in the significance spectrum,
3. a threshold for the cumulative significance related to a combined probability for all detected frequency components.

Projects

SigSpec is called by the command line

```
SigSpec <project>,
```

where `<project>` is the name of the SigSpec project. Before running the program, the user has to provide

1. a directory `<project>` used for the output,
2. a time series input file (see "Input", p. 4),

The project directory need not be empty, but SigSpec checks it for result files. The occurrence of a result file produces a warning message

```
Warning 009:  Result file found in project directory.
```

SigSpec appends new results to existing result files, whereas all other files are overwritten. If an old result file is present in the project directory, the results of the current SigSpec run may not be useful.

There are two conventions for denominating input files.

1. The standard method is to pass only one time series input file to the program. In this case, SigSpec expects the file to be named `<project>.dat`.

2. For an all-in-one analysis of multiple time series input files, i. e. for running SigSpec in MultiFile mode, a leading six-digit index is expected. In this case, the first file shall be named `000000.<project>.dat`, the next file is `000001.<project>.dat`, and so on. For more information on the MultiFile mode, please refer to "MultiFile mode", p. 21.

Furthermore, the user may pass a set of specifications to SigSpec by means of a file `<project>.ini` (see "Input", below). For specifications not given by the user, defaults are used.

Input

The input file for SigSpec is a time series. The only restriction to the format is that the number of items per row has to be constant for all rows in the file and that columns have to be separated by at least one whitespace character or tab. Dataset entries need not be numerical, except for the columns specified as time, observable, and weights (see "Special time series columns", p. 4).

An optional file `<project>.ini` consists of a set of keywords and arguments, defining project-specific parameters for SigSpec. If this file is not present, SigSpec uses a set of default parameters. A complete list of keywords is given in "Keywords Reference", p. 26.

Multiple use of the same keyword or the specification contradictory keywords causes the software to take into account only the last declaration. There are only two exceptions:

1. SigSpec accepts multiple weights columns specified by `col:weights`,
2. the simulator may be used to synthesize multiperiodic signal plus various types of noise upon the given sampling, where multiple use of the keywords `sim:signal`, `sim:poly`, `sim:exp`, `sim:zeromean`, `sim:serial`, `sim:temporal`, and `sim:rndsteps` is supported (see "Simulator Mode", p. 22).

SigSpec demands a carriage-return character at the end of the file `<project>.ini`, otherwise the program hangs!

Special time series columns
(keywords `col:time`, `col:obs`, `col:weights`)

The keywords `col:time` and `col:obs` determine those columns in the time series input file which contain time values and the observable monitored over

time, respectively. These columns are required and have to be uniquely specified. Column indices start with 1.

If `col:time` is not specified, the default value 1 is used. If `col:obs` is not specified, the default value 2 is used.

Furthermore, one or more columns with statistical weights may be specified using the keyword `col:weights`. The keyword accepts two arguments: the first is the column index, the second is a floating-point value, say p_n for the nth weights column. Given N weights columns indexed according to $n = 1, ..., N$, the total weight for the mth row is evaluated using the weights w_{nm} in the individual columns according to

$$\Gamma_m := \prod_{n=1}^{N} \gamma_{nm}^{p_n} \, . \tag{2}$$

Weights need not be normalised, this is performed by SigSpec.

Time, observable, and weights columns have to consist of floating-point numbers only. **The exponential annotation (e. g. `1.0385e-03`) is not supported!** SigSpec checks these columns before starting the computations. If a non-numeric entry is found in one of the special columns, the program terminates with one of the error messages

```
Error 010:  Time column format error.
Error 011:  Observable column format error.
Error 012:  Weights column format error.
```

The following pairs of keywords are designed for alternative use:

1. `ufreq` and `nycoef` determine the upper frequency limit for the spectra,
2. `freqspacing` and `osratio` determine the spacing between frequencies in the spectra.

If both keywords of a pair are determined, the first declaration is overwritten by the second one.

Lower frequency limit
(keyword `lfreq`)

The frequency where the computation of spectra starts is specified by the keyword `lfreq`. By default, the lower frequency limit is zero.

Upper frequency limit and Nyquist Coefficient
(keywords `ufreq`, `nycoef`)

The keyword `ufreq` allows to determine the upper limit of the frequency interval to be considered.

An alternative method is the automatic determination of this limit by means of the Nyquist Coefficient (keyword `nycoef`). For equidistantly sampled time series with sampling interval width δt, there is a uniquely defined *Nyquist Frequency*

$$f_\nu := \frac{1}{2\,\delta t}\,. \tag{3}$$

In case of non-equidistant sampling, each sampling interval between two consecutive time values may be considered to produce its individual Nyquist Frequency, whence this limit is ambiguous. In this case, the Nyquist Coefficient for an arbitrarily given frequency is introduced as the fraction of sampling intervals in the time domain the individual Nyquist Frequency of which is higher than the frequency under consideration. This provides to select an upper frequency limit by specifying a minimum Nyquist Coefficient. E. g., specifying a Nyquist Coefficient of 0.5 (which is the default value) guarantees that at least half of the information contained by the spectrum in the considered frequency range is below the Nyquist limit.

Frequency spacing and oversampling ratio
(keywords `freqspacing`, `osratio`)

The width of the interval between consecutive frequencies may be specified by the keyword `freqspacing`.

An alternative method is the automatic determination of the spacing by means of the oversampling ratio. In case of equidistantly sampled time series, the frequency spacing is defined by

$$\delta f := \frac{1}{\Delta t}\,, \tag{4}$$

where Δt denotes the width of the total time interval. For non-equidist time series, it is advisable to use a denser frequency sampling,

$$\delta f := \frac{1}{\Omega\,\Delta t}\,, \tag{5}$$

where Ω is the oversampling ratio. This quantity may be specified using the keyword `osratio`. The default value is 20, which is – in most cases – sufficient for practical use.

Program termination
(keywords `iterations`, `siglimit`), `csiglimit`)

There are three possibilities to specify a criterion for program termination.

1. The number of iterations may be provided using the keyword `iterations`.
2. A lower significance limit may be given by means of the keyword `siglimit`.
3. The reliability of the entire analysis is determined by the *cumulative significance*. It is based on the probability that at least one of the frequency components detected so far is due to noise. A threshold in terms of cumulative significance may be defined using the keyword `csiglimit`.

Multiple specifications in terms of these keywords cause the prewhitening cascade to terminate if one of the limits is reached.

The default assignment for `siglimit` is 5.4575. This pre-definition may be deactivated by defining

```
siglimit 0
```

in the `.ini` file. The alternative termination limits `iterations` and `csiglimit` are switched off by default.

Default Output

All output files are written into the project directory. A six-digit index denotes the iteration in the prewhitening cascade. E. g., an index `000000` represents a file obtained from the input data without any prewhitening, `000001` denotes a file after prewhitening of the first sinusoidal component. Furtherin, the general annotation `#iteration#` will be used for this six-digit identifier.

The keyword `output:spectra` may be used in the `.ini` file to restrict the number of spectra to be written to files.

Spectra

The filenames associated to the spectra are `s#iteration#.dat` with iteration indices starting at `000000`. The columns are

1. frequency [inverse time units],

2. significance,
3. amplitude [units of observable],
4. Fourier space phase angle [rad],
5. Fourier space phase angle of maximum significance [rad].

The phase angles θ are given according to a trigonometric fit,

$$F(t) := A\cos(2\pi ft - \theta) \ , \tag{6}$$

with amplitude A and frequency f as given in the file. This convention is compatible to the definition of phase in Fourier space. This definition of phase is consistently used for all types of SigSpec output.

Residual time series

After each prewhitening step, residuals are generated. The corresponding filenames are `r#iteration#.dat` with iteration indices starting at 000001. The column format is the same as for the time series input file. In addition, a file `residuals.dat` is generated representing the residuals after prewhitening all peaks found significant. If desired, the number of `r#iteration#.dat` files may be adjusted to the user's convenience using the keyword `output:residuals` in the `.ini` file. The file `residuals.dat` is not affected by this setting and generated at the end of the prewhitening sequence anyway.

The result file

The file `result.dat` contains a list of all identified significance maxima. The last entry in this file is associated to the last identified maximum, for which no prewhitening is performed any more. Similar to the preview, this file consists of six columns providing

1. frequency [inverse time units],
2. significance,
3. amplitude [units of observable],
4. phase [rad],
5. rms scatter of residuals,

6. point-to-point scatter of residuals,
7. the cumulative significance for all frequency components detected so far.

Specials

Spectral window

The spectral window is computed if the keyword `win` is given in the `.ini` file. This keyword does not require any parameters. The output is provided in the file `win.dat`. It consists of three columns referring to

1. frequency [inverse time units],
2. amplitude [units of observable],
3. Fourier-space phase angle [rad].

Preview

Since the prewhitening cascade performed by SigSpec may be extremely time consuming, the program can compute a preview. This add-on is activated by the keyword `preview` in the `.ini` file.

Whereas the significance spectra provide the False-Alarm Probability compared to a noise dataset with the same rms error as the given time series (or series of residuals, respectively), the significance spectrum provided in the file `preview.dat` represents a set of identified maxima in the significance spetrum of the original time series, but based on the point-to-point scatter in the time domain rather than the standard deviation of observables. The lower significance limit for writing a local maximum to the file `preview.dat` is specified as the argument to the keyword `preview` in the `.ini` file.

The calculation of significance is based on the assumption that only the point-to-point scatter is random and everything else contributing to the rms error represents signal that will be prewhitened in the course of the subsequent loop. The preview output is to be considered as a rough estimate for the final result obtained by step-by-step prewhitening and contains not only the intrinsic variations but also all aliases that are automatically omit in the following analysis. The file `preview.dat` consists of four columns referring to

1. frequency [inverse time units],
2. significance,

3. amplitude [units of observable],
4. phase [rad].

Sock Diagram

The Sock Diagram is an optional add-on of SigSpec. If the keyword `sock` is given in the `.ini` file, SigSpec computes significance levels for a constant signal-to-noise ratio at a set of different phase angles, and for all frequencies for which spectra are calculated. As described by Reegen (2006), the expected significance level for a given amplitude signal-to-noise ratio at constant frequency and phase angle is proportional to the signal-to-noise ratio. Significances in the Sock Diagram are normalised to an expected value of 1, corresponding to an amplitude signal-to-noise ratio

$$\frac{A}{\langle A \rangle} = \frac{2}{\sqrt{\pi \log e}} \approx 1.712\,. \tag{7}$$

The significance level for an arbitrary signal-to-noise ratio may be deduced by multiplying the significance displayed in the Sock Diagram by $\frac{\pi \log e}{4} \left(\frac{A}{\langle A \rangle}\right)^2 \approx 0.341 \left(\frac{A}{\langle A \rangle}\right)^2$.

Furthermore, the phase angle in the Sock Diagram is given with respect to the phase angle θ_0, i. e. the maxima in phase are consistently aligned to zero phase in order to avoid extreme variations of sock significance with frequency.

The corresponding output file `sock.dat` consists of three columns:

1. frequency [inverse time units],
2. phase with respect to significance maximum [rad],
3. significance.

The number of phase angles in the interval $[0, \pi[$ to be taken into account for each frequency of the spectrum has to be given as argument to the keyword `sock` in the `.ini` file. The significance levels in the phase intervals $[0, \pi[$ and $[\pi, 2\pi[$ are symmetric according to

$$\mathrm{sig}\,(A, \omega, \phi) = \mathrm{sig}\,(A, \omega, \phi + \pi)\ \forall \phi\,, \tag{8}$$

but both given in the output file `sock.dat` for completeness. The result represents a three-dimensional polar diagram of the sampling properties of the time

series input file. To enhance the corresponding plot resolution, the number of phases specified with the keyword `sock` in the `.ini` file is scaled by the maximum significance for each frequency. For significance maxima ≤ 1, the specified number is used, for significance maxima between 1 and 2, the number is doubled, and so on.

Example: screen output for time-unresolved analysis

In this example of the screen output generated by SigSpec, the project name is `test01`.

```
 SSSSSS  ii          SSSSSS
SS    SS            SS    SS
SS       ii  gggg g SS       p pppp   eeeee   ccccc
SS       ii gg   gg SS       pp   pp ee   ee cc   cc
 SSSSSS  ii gg   gg  SSSSSS  pp   pp ee   ee cc
      SS ii gg   gg       SS pp   pp eeeeeee cc
      SS ii gg   gg       SS pp   pp ee      cc
SS    SS ii gg   gg SS    SS pp   pp ee   ee cc   cc
 SSSSSS  ii  gggggg  SSSSSS  pppppp   eeeee   ccccc
                gg          pp
            gg   gg          pp
             ggggg           pp


SIGnificance SPECtrum
Version 1.2
************************************************************
by Piet Reegen
Institute of Astronomy
University of Vienna
Tuerkenschanzstrasse 17
1180 Vienna, Austria
Release date: February 12, 2006
```

The first task is to check whether the project directory is available.

```
*** command line interface *********************************
```

```
Checking availability of project directory test01...
Project directory test01 ok.
```

SigSpec initialises all parameters. Default values are assigned.

```
*** initialising parameters ********************************

time column index                        1
observable column index                  2
Nyquist coefficient                      0.5000000000000000
oversampling ratio                      20.0000000000000000
significance limit                       5.4575054153673648
```

Parameters are loaded from the file `test01.ini`, if present. SigSpec skips all lines that begin with #.

```
*** loading project-specific parameters ********************

number of weights columns                0
skipping                                           #test01:
upper frequency limit                   10.0000000000000000
spectral window
save sampling profile
preview: significance limit              5.4574999999999996
sock diagram: phases                    45
```

Here SigSpec checks whether the parameters specified by the user make sense.

```
*** checking parameters ************************************

pass
```

Now the program determines the number of columns and rows in the time series input file `test01.dat`. The width of the columns in the time series input file is used for formatted output of residual time series. Furthermore, SigSpec checks whether time and observable column contain numbers only. This check is also performed for weights columns. Then the time series is loaded from the input file. SigSpec returns the total number of entries.

```
*** checking time series input file ************************

time series                                 2 columns
                                          381 (non-empty) rows
column        1                            21 characters wide
column        2                            19 characters wide
numeric columns                             2
column        1                               numeric
column        2                               numeric
time column       (   1) ok
observable column (   2) ok
total number of entries read            762
```

The time series is sorted by ascending time. In this example, the input file is already sorted.

```
*** sorting time series by time ***************************

0 pairs exchanged.
```

Now the rms devation and the point-to-point scatter of the time series are displayed. The serial correlation between consecutive data points is evaluated.

```
*** time series statistics *********************************

time-domain rms deviation                 0.0044959150872315
time-domain point-to-point scatter        0.0035572749176717
time-domain serial correlation            0.3823955945138501
```

SigSpec sets interval limits for time-resolved analysis. In this example, time-resolved analysis has not been specified by the user. So there is not much to do in this step.

```
*** time-resolved analysis *********************************

preparing time series...
```

Here SigSpec displays relevant information concerning the spectra to be generated.

- In this example, the default value for the lower frequency limit is used.

- The frequency resolution is calculated as the inverse time interval width.
- The upper frequency limit has been specified in the file `test01.ini`.
- The Nyquist coefficient is calculated for the specified upper frequency limit.
- In this example, the default oversampling ratio is used.
- The frequency spacing is calculated for the specified oversampling ratio.
- The number of frequencies follows from the lower and upper frequency limit and the frequency spacing.

```
*** parameters for spectra *********************************

lower frequency limit                      0.0000000000000000
frequency resolution                       0.1089880382935977
upper frequency limit                     10.0000000000000000
Nyquist coefficient                        0.9736842105263158
oversampling ratio                        20.0000000000000000
frequency spacing                          0.0054494019146799
number of frequencies                   1836
```

The next step is the initialisation of the time interval. Since the time-resolved mode is deactivated, the total time interval is used here.

```
*** initialising time interval *****************************

time interval                                 0
from                                   2520.4365105400001994
to                                     2529.6118293700001232
start at row                                  1
stop at row                                 381
first time                             2520.4365105400001994
last time                              2529.6118293700001232
```

By default, SigSpec subtracts the mean value from all observable data. In this case, the mean value of `test01.dat` is already zero.

```
*** zero-mean correction ***********************************

correcting mean                            0.0000000000000000
```

The sampling profile is an essential part of the significance computation. All parameters to describe the influence of the time series sampling in Fourier space is entirely contained in the three parameters $/alpha_0$, $/beta_0$, and $/theta_0$, according to Reegen (2006). The sampling profile is always produced, but only written to a file if the keyword `profile` is provided in the `.ini` file, as in the present example.

```
*** sampling profile ***************************************

writing sampling profile to file test01/profile.dat
```

Now the spectral window is computed. This step is only performed if the keyword `win` is given in the `.ini` file.

```
*** spectral window ****************************************

writing spectral window to file test01/win.dat
```

The Sock Diagram is computed and written to a file, according to the keyword `sock` in the `.ini` file. If no keyword `sock` is given, this step is skipped.

```
*** knitting sock diagram **********************************

writing sock diagram to file test01/sock.dat
```

The point-to-point scatter of the observable column is evaluated and used for the significance computation that leads to the preview file. The number of local maxima found in the significance spectrum is returned. This is the number of lines in the file `preview.dat`.

```
*** generating preview *************************************

time-domain point-to-point scatter        0.0035572749176717
writing preview to file test01/preview.dat
number of local maxima                    12
```

Here follows the cascade of consecutive prewhitening. First, the significance spectrum of the original time series is generated. In this example, three prewhitening iterations are performed. For each iteration, SigSpec displays

- the index of the iteration, starting with 1,

- the rms deviation and point-to-point scatter of the current time series,
- the frequency, significance, amplitude, and phase for which the significance spectrum attains a maximum,
- the cumulative significance.

After each prewhitening step, the mean of the new time series is adjusted to zero. Then the residuals are written to the file `r#iteration#.dat` and the file `result.dat` is updated, if the conditions defined by the keywords `siglimit` and `csiglimit` in the `.ini` file. After terminating the prewhitening cascade, the file `residuals.dat` is generated.

```
*** running SigSpec ****************************************

writing spectrum to file test01/s000000.dat

iteration                                1
time-domain rms deviation                0.0044959150872315
time-domain point-to-point scatter       0.0035572749176717
frequency for maximum significance       3.1320510657673175
maximum significance                     9.5453895333947258
amplitude for maximum significance       0.0021389646409057
phase for maximum significance          -2.1173665771855146
cumulative significance                  9.5453895333947258
correcting mean                         -0.0001230415127870
writing residuals to file test01/r000000.dat
writing result to file test01/result.dat
writing spectrum to file test01/s000001.dat

iteration                                2
time-domain rms deviation                0.0042286090188032
time-domain point-to-point scatter       0.0035310867262463
frequency for maximum significance       3.9863491396163497
maximum significance                     7.4308483249462745
amplitude for maximum significance       0.0019073515661456
phase for maximum significance          -2.6025314442556748
cumulative significance                  7.4275249419990690
correcting mean                         -0.0002000642970939
writing residuals to file test01/r000001.dat
```

```
writing result to file test01/result.dat
writing spectrum to file test01/s000002.dat

iteration                                3
time-domain rms deviation                0.0040342412332152
time-domain point-to-point scatter       0.0035145549904548
frequency for maximum significance       5.4099700912143067
maximum significance                     5.0045652654576127
amplitude for maximum significance       0.0014003184431502
phase for maximum significance           2.2796051259411838
cumulative significance                  5.0029284426468950
writing residuals to file test01/residuals.dat

Finished.

************************************************************

Thank you for using SigSpec!
Questions or comments?
Please contact Piet Reegen (reegen@astro.univie.ac.at)
Bye!
```

Time-resolved Analysis

In time-resolved mode, SigSpec performs an analysis for a set of time intervals rather than the entire time series. An interval of width given by the keyword `timeres:range` is moved in steps of width given by the keyword `timeres:step` from the beginning of the time series to the end.[1] Time series data within such an interval are used to form a subset for which the analysis is performed. In addition, statistical weights may be applied to the subset data, all with respect to the center of the interval, which shall be denoted t_C.

The only exception is the weighting (keyword `timeres:w:damp`). In this case, the analysis is optimised for signal excited at the beginning of the time interval corresponding to the subset under consideration, t_B and exponentially damped towards the end of the subset.

The weight functions of time are given in Table 1. Normalisation of weights

[1] In general, the step width is slightly modified by the software to achieve time-resolved analysis over the entire time series.

keyword	arguments	weight function
`timeres:w:none`		1
`timeres:w:ipow`	ξ	0 if $t = t_C$, $\lvert t - t_C \rvert^{-\xi}$ else
`timeres:w:gauss`	σ	$\mathrm{e}^{-\left(\frac{t-t_C}{\sigma}\right)^2}$
`timeres:w:exp`	ζ	$\mathrm{e}^{-\frac{\lvert t-t_C \rvert}{\zeta}}$
`timeres:w:damp`	ζ	$\mathrm{e}^{-\frac{t-t_B}{\zeta}}$
`timeres:w:cos`	ν, Φ	$\cos\left(2\pi\nu \lvert t - t_C \rvert - \Phi\right)$
`timeres:w:cosp`	ν, Φ, ξ	$\cos^{\xi}\left(2\pi\nu \lvert t - t_C \rvert - \Phi\right)$

Table 1: Weight functions for time-resolved SigSpec analysis. The beginning of the time interval associated with the referring subset is denoted t_B, whereas t_C symbolises the center of the time interval.

is automatically performed by SigSpec.

In time-resolved mode, the set of output files as given in Chapter is generated for each subset of the time series. This requires the introduction of a second six-digit index, `#interval#`, in addition to `#iteration#`, and the annotation of output files is

1. `wts.#interval#.dat` for the weight function vs. time in each subset,
2. `s#iteration#.#interval#.dat` for the spectra,
3. `r#iteration#.#interval#.dat` for the residuals after each step of prewhitening,
4. `result.#interval#.dat` for the result file with a list of prewhitened signal components,
5. `residuals.#interval#.dat` for the final residuals after prewhitening all significant signal components.

The column syntax is strictly consistent with the time-unresolved versions (see Chapter). The additional files, `wts.#interval#.dat`, are in two-column format. The first column represents the time values in the corresponding subset, the second column contains the weight function without normalisation.

Special functions – as introduced in "Specials", p. 9) – are also supplied with the `#interval#` index, i. e.

1. `win.#interval#.dat` for the amplitude window,
2. `profile.#interval#.dat` for the sampling profile,
3. `preview.#interval#.dat` for the preview,
4. `sock.#interval#.dat` for the Sock Diagrams.

Note that the latter use individual time axis zero-points for all intervals.

Example: screen output for time-resolved analysis

In this section, an example of the screen output generated for time-resolved SigSpec analysis is given. The example uses `test09` as the project name. Those parts of the output which are similar to the example given in "Example: screen output for time-unresolved analysis" (p. 11) are skipped.

The project-specific parameters also contain the specification of the time-resolved mode. The number of iterations for each time interval is set 1, whence no prewhitening is performed, and only the "original" spectra are computed. A rectangular filter is used as the weight function.

```
*** loading project-specific parameters ********************

number of weights columns                 0
skipping                                                  #
number of iterations                      1
range (time-resolved analysis)            0.8000000000000000
step width (time-resolved analysis)       1.0000000000000000
time-resolved analysis mode                   moving average
```

SigSpec prepares the time series for time-resolved analysis. First, the number of time intervals, i. e. subsets, is computed. Based on this number, the program re-adjusts the step width to achieve coverage of the entire time series.

```
*** time-resolved analysis *********************************

preparing time series...
number of time intervals                  9
time interval width                       0.8000000000000000
time step width                           1.0469148537499904
```

The prewhitening loop starts with the generation of a time series subset.

```
*** initialising time interval ******************************

time interval                              0
from                                2520.4365105400001994
to                                  2521.2365105400003813
start at row                               1
stop at row                               24
first time                          2520.4365105400001994
last time                           2521.3497449500000585
```

The (Gaussian) weight function is applied to the time series subset and saved.

```
*** generating weights ***************************************

writing weights to file test09/wts000000.dat
```

The adjustment of the mean value of the observable data is performed for each subset.

```
*** zero-mean correction (time-resolved) ********************

correcting mean                             0.0007278824541693
```

The sampling profile is computed for each subset. The screen output of the significance computation is similar to the time-unresolved version. In this example, no significant signal component is detected in the given subset.

```
*** running SigSpec ****************************************

writing spectrum to file test09/s000000.000000.dat

iteration                                   1
time-domain rms deviation                   0.0044959150872315
time-domain point-to-point scatter          0.0035572749176717
frequency for maximum significance         19.3600680207706901
maximum significance                        2.6659071825608560
amplitude for maximum significance          0.0042356896121601
```

```
phase for maximum significance          -0.2291457395195020
cumulative significance                  2.6659071825608560
writing residuals to file test09/residuals000000.dat
```

After finishing the prewhitening cascade for a subset (in this case, the cascade consists of one procedure only), the observable column has to be restored to return to the original values.

```
*** reconstructing time series ******************************
```

MultiFile Mode

An additional feature of SigSpec is the ability to handle multiple time series input files at once. This increases the performance of the program significantly, if the time values in all input files are identical.

- The user has to provide only one project directory `<project>` – just as in standard mode.
- Parameter specifications in the file `<project>.ini` have to be unique for all time series input files. Thus SigSpec expects the same column format for all time series input files and applies the settings specified in the `.ini` file to all input files.
- Time series input files have to be indexed according to `#multifile#.<project>.dat`, where `#multifile#` represents a six-digit index starting with `000000`. Note that strictly ascending indices are required.
- All output files are supplied with the leading index `#multifile#`. For example, `000012.s000009.000002` denotes the significance specturm the second iteration for time interval number nine in a time-resolved analysis of the 12th file in MultiFile mode.
- The MultiFile mode is activated by the keyword `multifile`, followed by an integer value. This value is interpreted as the maximum index up to which the calculations shall be performed. This permits a restriction for, e. g., test runs. If the index limit is assigned a negative value, SigSpec analyses as many files as available.
- The sampling profile of the file `000000.<project>.dat` is always written to a file. For subsequent and consistent time series, the sampling profile

is taken from this file, which saves computation time. Only if the new time values are inconsistent with those of the precursor, the profile is re-calculated and stored in a corresponding output file for later use.

The Built-in Simulator

SigSpec contains a simulator to generate and analyse synthetic time series. To activate the simulator, a sequence of keywords may be given in the `.ini` file to generate a variety of datasets. The sampling is taken from the time series input file.

The simulator activities specified by the keywords `sim:signal`, `sim:poly`, `sim:exp`, `sim:serial`, `sim:temporal`, `sim:rndsteps`, and `sim:zeromean` are interpreted as a sequence and performed step by step, following their order in the `.ini` file. SigSpec generates the synthetic light curve by adding up all activities in the defined order.

The simulator mode
(keywords `sim:off`, `sim:add`, `sim:replace`)

SigSpec supports two different simulator modes.

1. The keyword `sim:add` runs the simulator in additive mode. The program keeps the original observable values and adds the synthetic values. For example, this function is useful to add synthetic noise to a given time series.
2. The keyword `sim:replace` forces the simulator to overwrite the original observable values with the synthetic values.
3. The keyword `sim:off` is the default setting and need not explicitly be specified if no simulator activity is desired.

Sinusoidal signal
(keyword `sim:signal`)

The keyword `sim:signal` is given with five floating-point parameters. They specify

1. the lower time limit,
2. the upper time limit,

3. the amplitude,
4. the time zeropoint (a fixed time where the signal shall attain a maximum), and
5. the frequency [inverse time units].

If the lower and upper time limits are both set zero, the signal is generated for the entire time base.

Polynomial trend
(keyword `sim:poly`)

The keyword `sim:poly` is given with five floating-point parameters. They specify

1. the lower time limit,
2. the upper time limit,
3. the coefficient P_0,
4. the time zeropoint t_0, and
5. the exponent X.

The polynomial trend is generated by the relation

$$P\left(t\right) := P_0 \left(t - t_0\right)^X \ . \tag{9}$$

If the lower and upper time limits are both set zero, the polynomial trend is generated for the entire time base.

Exponential trend
(keyword `sim:exp`)

The keyword `sim:exp` is given with five floating-point parameters. They specify

1. the lower time limit,
2. the upper time limit,
3. the coefficient E_0,
4. the time zeropoint t_0, and

5. the exponent X.

The polynomial trend is generated by the relation

$$E\left(t\right) := E_0\, \mathrm{e}^{X(t-t_0)}\ . \tag{10}$$

If the lower and upper time limits are both set zero, the exponential trend is generated for the entire time base.

Serially correlated noise
(keyword `sim:serial`)

This simulator module produces Gaussian noise the standard deviation of which may vary in time according to a polynomial trend. A serial correlation coefficient between consecutive data points may be specified.

The keyword `sim:serial` is given with six floating-point parameters. They specify

1. the lower time limit,
2. the upper time limit,
3. the coefficient σ_0 for the standard deviation of the Gaussian noise,
4. the time zeropoint t_0 for the polynomial trend of the standard deviation,
5. the exponent X for the polynomial trend of the standard deviation, and
6. the serial correlation coefficient.

The standard deviation of the Gaussian noise follows the relation

$$\sigma\left(t\right) := \sigma_0\ \left(t - t_0\right)^X\ . \tag{11}$$

If the lower and upper time limits are both set zero, the noise is generated for the entire time base.

Temporally correlated noise
(keyword `sim:temporal`)

This simulator module produces Gaussian noise the standard deviation of which may vary in time according to a polynomial trend. A temporal correlation coefficient R_T between consecutive data points t_{n-1}, t_n may be specified. In

contrary to the serial correlation, the temporal correlation takes into account the width of the time interval between pairs of data points, which has implications on the noise behaviour of non-equidistantly sampled data. The serial correlation R_S drops exponentially with the distance in time according to

$$R_S := R_T^{t_n - t_{n-1}} \; . \tag{12}$$

In this context, the temporal correlation coefficient may be interpreted as the serial correlation coefficient of two data points separated by one unit of time.

The keyword `sim:temporal` is given with six floating-point parameters. They specify

1. the lower time limit,
2. the upper time limit,
3. the coefficient σ_0 for the standard deviation of the Gaussian noise,
4. the time zeropoint t_0 for the polynomial trend of the standard deviation,
5. the exponent X for the polynomial trend of the standard deviation, and
6. the temporal correlation coefficient R_T.

The standard deviation of the Gaussian noise follows the relation

$$\sigma\left(t\right) := \sigma_0 \; \left(t - t_0\right)^X \; . \tag{13}$$

If the lower and upper time limits are both set zero, the noise is generated for the entire time base.

Random steps
(keyword `sim:rndsteps`)

This module generates steps following two random processes:

1. the constant attained by the synthetic observable throughout each step follows a Gaussian distribution with an expected value of zero,
2. a Poisson process is used to define when a step has to be incorporated.

The keyword `sim:rndsteps` is given with four floating-point parameters. They specify

1. the lower time limit,
2. the upper time limit,
3. the standard deviation of the Gaussian distribution defining the constants attained throughout each step,
4. the expected time range for the Poisson distribution of steps.

If the lower and upper time limits are both set zero, the steps are generated for the entire time base.

Zero-mean adjustment
(keyword `sim:zeromean`)

The keyword `sim:zeromean` may be used to adjust the mean value of the time series (or a subset) to zero. It is given with two floating-point parameters,

1. the lower time limit, and
2. the upper time limit.

If the lower and upper time limits are both set zero, the mean value of the entire synthetic time series is adjusted to zero.

Keywords Reference

This section is a compilation of all keywords accepted by SigSpec. A brief description of arguments and default values is given. The type of argument is provided by either `<int>` or `<double>`, and default values are given in brackets, e. g. `(5.4575)`. Empty brackets indicate that there is no default setting.

Time series

`col:time <int> (1)`

time column index (unique), starting with 1

`col:obs <int> (2)`

observable column index (unique), starting with 1

`col:weights <int> ()`

weights column (also multiple), starting with 1

Spectra

`lfreq <double> (0)`

lower frequency limit [inverse time units]

`ufreq <double> ()`

upper frequency limit [inverse time units]

`nycoef <double> (0.5)`

Nyquist coefficient

`freqspacing <double> ()`

spacing between consecutive frequencies [inverse time units]

`osratio <double> (20)`

oversampling ratio

Program termination

`iterations <int> ()`

number of prewhitening iterations

`siglimit <double> (5.4575)`

lower significance limit (0 to deactivate)

`csiglimit <double> ()`

lower cumulative significance limit

Time-resolved analysis

`timeres:range <double> ()`

subset interval width [time units]

`timeres:step <double> ()`

step width between subset centres [time units]

`timeres:w:none`

unweighted moving averages

`timeres:w:ipow <double> ()`

inverse power weights
parameter: exponent

`timeres:w:gauss <double> ()`

Gaussian weights
parameter: standard deviation [time units]

`timeres:w:exp <double> ()`

exponential weights
parameter: width [time units]

`timeres:w:damp <double> ()`

exponential damping
parameter: width [time units]

`timeres:w:cos <double> <double> ()`

cosine weights
parameters:

- frequency [inverse time units]
- phase [rad]

`timeres:w:cosp <double> <double> <double> ()`

weights according to the power of a cosine
parameters:

- frequency [inverse time units]
- phase [rad]
- exponent

Simulator

`sim:off`

deactivate simulator (default)

`sim:add`

add synthetic data to given observable

`sim:replace`

replace given observable by synthetic data

`sim:signal <double> <double> <double> <double> <double> ()`

sinusoidal signal
parameters:

- lower time limit [time units]
- upper time limit [time units]
- amplitude
- time zeropoint [time units]
- frequency [inverse time units]

`sim:poly <double> <double> <double> <double> <double> ()`

polynomial trend
parameters:

- lower time limit [time units]
- upper time limit [time units]
- scale
- time zeropoint [time units]
- exponent

`sim:exp <double> <double> <double> <double> <double> ()`

exponential trend
parameters:

- lower time limit [time units]
- upper time limit [time units]
- scale
- time zeropoint [time units]
- exponent

`sim:serial <double> <double> <double> <double> <double> <double> ()`

serially correlated noise
parameters:

- lower time limit [time units]
- upper time limit [time units]
- scale for standard deviation
- time zeropoint for polynomial trend of standard deviation [time units]
- exponent for polynomial trend of standard deviation
- serial correlation coefficient

`sim:temporal <double> <double> <double> <double> <double> <double> ()`

temporally correlated noise
parameters:

- lower time limit [time units]
- upper time limit [time units]
- scale for standard deviation
- time zeropoint for polynomial trend of standard deviation [time units]
- exponent for polynomial trend of standard deviation
- temporal correlation coefficient

`sim:rndsteps <double> <double> <double> <double> ()`

random steps
parameters:

- lower time limit [time units]
- upper time limit [time units]
- standard deviation for Gaussian distribution of (constant) step values
- expected time range for Poisson distribution of steps [time units]

`sim:zeromean <double> <double> ()`

zero-mean adjustment
parameters:

- lower time limit [time units]
- upper time limit [time units]

MultiFile mode

`multifile <int> ()`

activate MultiFile mode
parameter: maximum number of time series input files (≤ 0 ... infinite)

Error Messages and Warnings

`Error 001: Command line syntax.`

The appropriate command to run SigSpec is

`SigSpec <project>.`

This error message indicates that the project name has not been specified. SigSpec accepts only one command-line argument and ignores additional tags.

`Error 002: File not found.`

The name time series input file is the project name, followed by the extension `.dat`. In MultiFile mode, the program checks the availability of the first indexed time series input file, `000000.<project>.dat`. This error is returned if no file with the given name is found.

`Error 003: Failed to count weights column specifiers.`

In the course of loading the `.ini` file, SigSpec determines the number of specified weights columns (i. e. the number of keywords `col:weights`). This error message indicates that the program is not able to perform counting, e. g. due to a format error in the `.ini` file.

`Error 004: Illegal parameter, value >= 0 required.`

The lower frequency limit (keyword `lfreq`) has to be non-negative.

`Error 005: Illegal parameter, value > 0 required.`

SigSpec expects the following parameters to attain strictly positive values:

- the upper frequency limit (keyword `ufreq`),
- the frequency spacing (keyword `freqspacing`),
- the oversampling ratio (keyword `osratio`),
- the number of iterations (keyword `iterations`),
- the cumulative significance limit (keyword `csiglimit`),

- the subset interval width (keyword `timeres:range`) for time-resolved analysis,

- the step width (keyword `timeres:step`) for time-resolved analysis.

If the calculated number of subsets for time-resolved analysis is non-positive, this error message is returned as well.

```
Error 006:  Illegal parameter, value >= 0 and <= 1 required.
```

This error message is returned if an illegal value has been specified for the Nyquist coefficient (keyword `nycoef`).

```
Error 007:  Illegal column index, value > 0 required.
```

Column indices start with 1. If either the time or the observable column are specified non-positive (keywords `col:time`, `col:obs`), this error message is returned. If non-positive weights column indices are specified (keyword `col:weights`), SigSpec ignores the corresponding declaration.

```
Error 008:  Project directory not found.
```

The project name is provided in the command line. SigSpec checks whether a project directory with the specified name is available. If not, this error message is returned.

```
Warning 009:  Result file found in project directory.
```

This warning message indicates a non-empty project directory. In case of an old result file in this directory, SigSpec appends the results of the current run to the end of the existing result file. The user is warned that the output may not be useful.

This applies to time-unresolved as well as time-resolved mode.

```
Error 010:  Time column format error.
```

If the column specified as time (keyword `col:time` contains non-numeric entries, SigSpec terminates with this error message.

`Error 011: Observable column format error.`

If the column specified as observable (keyword `col:obs` contains non-numeric entries, SigSpec terminates with this error message.

`Error 012: Weights column format error.`

If a column specified as statistical weights (keyword `col:weights` contains non-numeric entries, SigSpec terminates with this error message.

`Error 013: Multiply assigned column index.`

The assignment of time, observable, and weights columns has to be unique (keywords `col:time`, `col:obs`, `col:weights`). If a column index is assigned to two different types, SigSpec terminates with this error message.

`Error 014: Cannot read time series.`

This error message is returned if the attempt to load the time series fails.

`Error 015: Time series has < 2 data points.`

Fourier and significance analysis does not make sense for a single data point. If the time series input file is empty or has only one row, SigSpec terminates with this error message.

`Error 016: Sorting failed.`

SigSpec automatically sorts the time series by ascending time. If the sorting procedure fails, this error message is returned.

`Error 017: Failed to open output file.`

This error message indicates that the creation of an output file has failed, e. g. due to inappropriate permission settings.

`Error 018: Illegal parameter for time-resolved analysis.`

Table 1 (p. 18) contains a summary of weight functions available for time-resolved analysis. If the specification of parameters is illegal or incomplete, SigSpec terminates with this error message.

`Warning 019: Skipping time interval with < 2 data points.`

If a subset interval in time-resolved mode is empty or contains only a single data point, the significance computation is skipped for this subset. To indicate the occurrence of such a degenerate subset, SigSpec returns this warning.

`Error 020: Simulator error in signal generator.`

This error message indicates an illegal parameter specification for the keyword `sim:signal`.

`Error 021: Simulator error in polynomial trend generator.`

This error message indicates an illegal parameter specification for the keyword `sim:poly`.

`Error 022: Simulator error in exponential trend generator.`

This error message indicates an illegal parameter specification for the keyword `sim:exp`.

`Error 023: Simulator error in serial correlation generator.`

This error message indicates an illegal parameter specification for the keyword `sim:serial`.

`Error 024: Simulator error in temporal correlation generator.`

This error message indicates an illegal parameter specification for the keyword `sim:temporal`.

`Error 025: Simulator error in random steps generator.`

This error message indicates an illegal parameter specification for the keyword `sim:rndsteps`.

`Warning 026: Inconsistent number of time series columns.`

This warning message is returned in MultiFile mode, indicating that the current time series has not got the same number of columns as its precursor.

`Warning 027: Inconsistent number of time series rows.`

This warning message is returned in MultiFile mode, indicating that the current time series has not got the same number of rows as its precursor.

`Warning 028: Inconsistent number of numeric time series columns.`

This warning message is returned in MultiFile mode, indicating that the current time series has not got the same number of purely numeric columns as its precursor.

`Warning 029: Inconsistent time value detected.`

This warning message is returned in MultiFile mode, that a time value in the current time series differs from the corresponding time value of the previous time series.

`Warning 030: Inconsistent weight detected.`

This warning message is returned in MultiFile mode, that a weight in the current time series differs from the corresponding weight of the previous time series.

`Error 031: Unexpected end of profile file.`

This warning message is returned in MultiFile mode, indicating that the profile file was corrupted, i. e. the length of the stored profile does not match the length of the required profile.

Online availability

SigSpec is available online at
`ams.astro.univie.ac.at/computer/SigSpec/SigSpecHome.html`.
There are builds for various operating systems and CPUs available for free download. The ANSI-C code is available on request. Please contact P. Reegen (`reegen@astro.univie.ac.at`).

Acknowledgments. It is a pleasure to thank T. Appourchaux (IAS, Orsay), T. Boehm (Obs. M.-P., Toulouse), A. Baglin (Obs. Paris, Meudon), M. Breger, R. Dvorak, M. G. Firneis, D. Frast (Univ. Vienna), R. Garrido

(Inst. Astrof. Andalucia, Granada), M. Gruberbauer (Univ. Vienna), D. B. Guenther (St. Mary's Univ., Halifax), M. Hareter, D. Huber, T. Kallinger (Univ. Vienna), R. Kuschnig (UBC, Vancouver), S. Marchenko (Western Kentucky Univ., Bowling Green, KY), M. Masser (Univ. Vienna), J. M. Matthews (UBC, Vancouver), E. Michel (Obs. Paris, Meudon), A. F. J. Moffat (Univ. Montreal), D. Punz (Univ. Vienna), V. Ripepi (INAF, Naples), S. M. Rucinski (D. Dunlap Obs., Toronto), T. A. Ryabchikova (Inst. Astr. RAS, Moscow), D. Sasselov (Harvard-Smithsonian Center, Cambridge, MA), S. Schraml (Univ. Vienna), G. A. Wade (RMC, Kingston), G. A. H. Walker (UBC, Vancouver), W. W. Weiss, and K. Zwintz (Univ. Vienna) for valuable discussion and support with extensive software tests, respectively.

References

Breger, M., Stich, J., Garrido, R., Martin, B., Jiang S.-Y., Li Z.-P., Hube, D.P., Ostermann, W., Paparo, M., Scheck, M. 1993, A&A, 271, 482
Reegen, P. 2005a, in *The A-Star Puzzle*, Proceedings of IAUS 224, eds. J. Zverko, J. Ziznovsky, S.J. Adelman, W.W. Weiss (Cambridge: Cambridge Univ. Press), p. 791
Reegen, P. 2005b, ApJ, in preparation
Scargle, J.D. 1982, ApJ, 263, 835